Row-Ordering Schemes for Sparse Givens Transformations. III. Analyses for a Model Problem*

Alan George
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Joseph Liu
Department of Computer Science
York University
Downsview, Ontario, Canada M3J 1P3

and

Esmond Ng
Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Submitted by Richard A. Brualdi

ABSTRACT

In [7, 8], we showed that a good row ordering can be obtained from a width-1 nested-dissection column ordering in the orthogonal decomposition of sparse matrices using Givens transformations. The objective of this paper is to analyze a model $k \times k$ grid problem and to show that the number of multiplicative operations required to transform the sparse rectangular matrix associated with the model problem is $O(k^3)$ if width-1-nested dissection column ordering and the associated good row ordering are used. We also demonstrate that if the column ordering is a (width-1 or width-2) nested-dissection ordering, there exists a row ordering such that the cost of the computation is at least $O(k^4)$.

*The research was sponsored by the Canadian Natural Sciences and Engineering Research Council under Grants A5509 and A8111, and was also sponsored by the Applied Mathematical Sciences Research Program, Office of Energy Research, U.S. Department of Energy under contract DE-AC05-84OR21400 with the Martin Marietta Energy Systems Inc., and by the U.S. Air Force Office of Scientific Research under contract AFOSR-ISSA-84-00056.

52 Vanderbilt Ave., New York, NY 10017 0024-3795/86/$3.50

1. INTRODUCTION

In the QR decomposition of sparse matrices using Givens transformations, George and Ng have shown that a good row ordering exists if the column ordering is a width-2 nested-dissection ordering [9]. If we define the *first* (*last*) *subscript* of a row to be the column subscript of the first (last) nonzero in that row, then a good row ordering for a width-2 nested-dissection column ordering, which will be referred to as the row ordering *induced* by a width-2 nested-dissection column ordering, can be obtained by arranging the rows of the (column) permuted matrix so that the first subscripts are in ascending order. Then in [7, 8], we introduced two graph models to study the row-ordering problem. We showed that if a width-1 nested-dissection column ordering is used, then a good row ordering, which will be referred to as the row ordering *induced* by a width-1 nested-dissection column ordering, can also be obtained by arranging the rows of the (column) permuted matrix so that the last subscripts are in ascending order.

The objective of this paper is to analyze a model problem using the two graph models introduced in [7, 8] and to demonstrate that the row ordering induced by a width-1 nested-dissection column ordering is as good as the one induced by a width-2 nested-dissection column ordering. We assume the reader is familiar with the results obtained in [7] and [8].

Consider a model $k \times k$ grid problem which is typical in the natural factor formulation of finite-element methods [1, 2, 9]. Let A denote the rectangular matrix obtained in the model problem. In [9], it was shown that if a width-2 nested-dissection column ordering and its induced row ordering are used, the number of multiplicative operations required in computing the QR decomposition using rotations is $O(k^3)$. We consider width-1 nested-dissection column ordering in this paper. Assume the columns of A are labeled by the width-1 nested dissection algorithm. We show that if the induced row ordering is used, the number of multiplicative operations required in computing the decomposition is also $O(k^3)$. We also show that the row-ordering problem is important by providing a row ordering such that the number of multiplicative operations required in computing the QR decomposition is at least $O(k^4)$ even when a (width-1 or width-2) nested dissection column ordering is used.

An outline of the paper is as follows. In Section 2, we define the model problem and describe the bipartite graph and symmetric graphs associated with it. We then review nested dissection for the model problem in Section 3. In Section 4, we use the graph models to analyze the complexity of computing the QR decomposition for the model problem when a width-1 nested-dissection column ordering and its induced row ordering are used. A "bad" row ordering for (width-1 or width-2) nested-dissection column orderings is described in Section 5. Finally, some concluding remarks are provided in Section 6.

Related work on the row-ordering problem can be found in [3, 4].

2. A MODEL PROBLEM

In the following sections, we consider the width-1 nested-dissection algorithm for finding good row and column orderings in the orthogonal decomposition of sparse rectangular matrices using Givens transformations. The complexity of the algorithm will be analyzed for a model problem. This model problem is defined on a $k \times k$ grid, and it is typical of those arising in the natural factor formulation of finite-element methods [1, 2, 9].

Consider a $k \times k$ grid which consists of $(k-1)^2$ small squares (or elements). An example is given in Figure 1 with $k = 4$. For our purpose, the model problem is defined as follows. Associated with each of the k^2 grid points (or nodes) is a variable, and associated with each of the $(k-1)^2$ small squares is a set of four equations (or rows) involving the four variables at the corners of the square. This gives rise to a large sparse overdetermined system of linear equations with $4(k-1)^2$ equations and k^2 unknowns which are the variables at the grid points. We denote the coefficient matrix by A. The matrix associated with the example in Figure 1 is shown in Figure 2. The matrix A is reduced to upper trapezoidal form using Givens transformations. Let R denote the $k^2 \times k^2$ upper triangular matrix obtained after the transformation.

In [7, 8], we have presented two graph models to study the row-ordering problem in the QR decomposition of sparse matrices using rotations: an explicit model that uses bipartite graphs, and an implicit model that employs

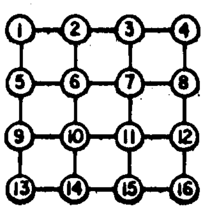

FIG. 1. A 4×4 finite-element grid.

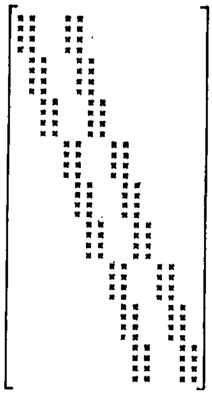

FIG. 2. The matrix associated with a 4×4 finite-element grid.

symmetric graphs of $(a^i)^T(a^i)$, where a^i denotes the ith row of A. In the remainder of this section, we describe the bipartite graph and symmetric graph associated with the model problem. The notation is the same as that used in [7, 8].

We first consider the bipartite graph of A, which we denote by $H(A) = (Q(A), X(A), B(A))$. Note that all four equations associated with a small square in the $k \times k$ grid have the same structure. Thus in the bipartite graph, there will be four vertices in $Q(A)$ whose adjacent sets in $H(A)$ are identical. In order to make the presentation cleaner, these four (row) vertices are "collapsed" into a single one in the figures shown in this paper. The row labeling has the form $\{4, 8, 12, \ldots, 4j, \ldots\}$. This is to be interpreted as follows: a (collapsed) row vertex labelled $4j$ represents rows $4j-3$, $4j-2$, $4j-1$, and $4j$. The bipartite graph associated with the example in Figures 1 and 2 is given in Figure 3, in which circles and boxes correspond respectively to the column and (collapsed) row vertices.

The symmetric graph associated with A is simply the symmetric graph of $M = A^TA$, which we denote by $G(M) = (X(M), E(M))$. [Of course, $X(M) = X(A)$.] Note that

$$A^TA = \sum_i (a^i)^T(a^i),$$

and clearly $(a^i)^T(a^i)$ contains a 4×4 dense matrix involving the variables associated with a small square in the grid. Thus the symmetric graph is identical to the $k \times k$ grid, except that the four grid points at the corners of each small square are now pairwise connected. The example shown in Figure

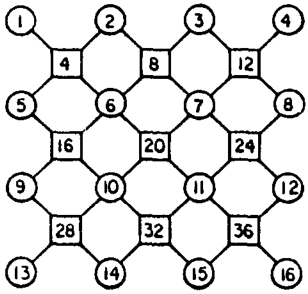

FIG. 3. Bipartite graph of the matrix shown in Figure 2.

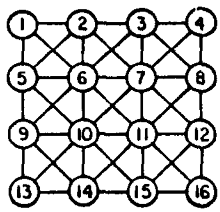

FIG. 4. Symmetric graph associated with the matrix shown in Figure 2.

4 is the symmetric graph associated with the example given in Figures 1 and 2.

Throughout this paper, the terms "grid points", "column vertices" (column nodes) in the bipartite graph $H(A)$, and "vertices" (nodes) in the symmetric graph $G(M)$ will be used interchangeably.

3. NESTED DISSECTION FOR THE MODEL PROBLEM

We will use the symmetric graph $G(M)$ in this section, since it is more convenient to use in describing nested dissection for $k \times k$ grids. If the nodes on a set of consecutive (horizontal or vertical) grid lines, together with the edges incident from them, are removed, the remaining graph will be disconnected. Such a set of nodes forms a *separator*. A separator is a *width-w separator* if w consecutive grid lines are removed. In this paper, we are interested in width-1 and width-2 separators, which are illustrated in Figures 5 and 6.

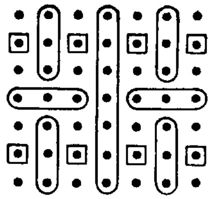

FIG. 5. Width-1 nested-dissection.

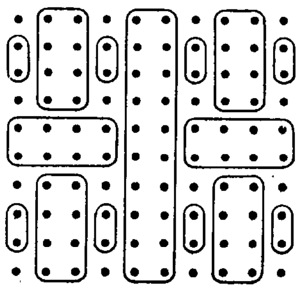

FIG. 6. Width-2 nested dissection.

To preserve symmetry, we assume the width-w separator is chosen from the "center" of the graph. This separator will be called a *level-1 separator*. The remaining graph then consists of two disconnected subgraphs, each of which is a symmetric graph associated with an (approximately) $k \times k/2$ rectangular grid. We can apply this *dissection* technique to each of these two subgraphs, yielding four subgraphs, each of which is now a symmetric graph associated with an (approximately) $k/2 \times k/2$ grid. The separators are called *level-2 separators*.

This technique can of course be applied recursively to each subgraph until there are no separators in the remaining subgraphs, and this yields a *width-w nested dissection*. Examples illustrating width-1 and width-2 nested dissection are given in Figures 5 and 6. Note that the dissection strategy effectively partitions the grid points into disjoint sets. Such a partitioning is referred to as a *width-w nested-dissection partitioning*.

Lemma 3.1 is an immediate consequence of the dissection technique.

LEMMA 3.1.

(1) *The number of levels of dissections is approximately* $2\log_2(k)$.

(2) *The number of level-h separators is approximately* 2^{h-1}, *for* $1 \leqslant h \leqslant 2\log_2(k)$.

(3) *The number of grid points on a level-h width-w separator is approximately* $2^{-\lfloor h/2 \rfloor}kw$, *for* $1 \leqslant h \leqslant 2\log_2(k)$.

In the lemma, *approximately* means that the formula referred to is in error by lower-order terms. We adopt this imprecision in order to keep the expressions simple and to avoid clouding the essential point of the lemma through including the numerous additional low-order terms that would be necessary to make the statements precise.

Consider relabeling the grid points in a $k \times k$ grid [that is, the nodes of $X(M)$ and the columns of A]. Assume we have determined a nested-dissection partitioning. Let S denote a level-1 separator. Suppose $x \in \mathrm{S}$ and $y \notin \mathrm{S}$. Then we will label x after y. The same labeling strategy is applied recursively to the nodes in the set $X(M) - \mathrm{S}$. The resulting ordering is called a *nested-dissection (vertex) ordering*. We denote the vertex in $X(M)$ [and $X(A)$] having label i by x_i.

4. WIDTH-1 NESTED-DISSECTION ROW ORDERING

In [7, 8], we have shown that a good row ordering exists if a width-1 nested-dissection column ordering is used for A. This "induced" row ordering is obtained by arranging the rows of the (column) permuted matrix so that the last subscripts are in ascending order. In this section we use both the bipartite-graph model and the implicit graph model to show that, for the model problem, the number of multiplicative operations required to transform A to upper trapezoidal form using rotations is $O(k^3)$ if width-1 nested-dissection column ordering and its induced row ordering are used. In the following discussion, $S_h \subseteq X(M)$ will denote a level-h width-1 separator.

4.1. *Bipartite-Graph Model*

Assume that we have obtained a width-1 nested-dissection ordering for the column vertices $X(A)$. The induced row ordering can be described as follows. Let $q \in \mathrm{Adj}_{H(A)}(\mathrm{S}_1)$ and $\bar{q} \notin \mathrm{Adj}_{H(A)}(\mathrm{S}_1)$. Suppose s and t are respectively the last subscripts of the rows associated with q and $\bar{q}$. The labeling of the column vertices implies that $x_s \in \mathrm{S}_1$ and $x_t \notin \mathrm{S}_1$. Hence q will be labeled *after* $\bar{q}$, since $s > t$. In other words, the set $\mathrm{Adj}_{H(A)}(\mathrm{S}_1)$ contains the row vertices that are labeled *last*. The same argument is applied recursively to label the row vertices remaining in $Q(A) - \mathrm{Adj}_{H(A)}(\mathrm{S}_1)$. An example illustrating width-1 column ordering and its induced row ordering is given in Figure

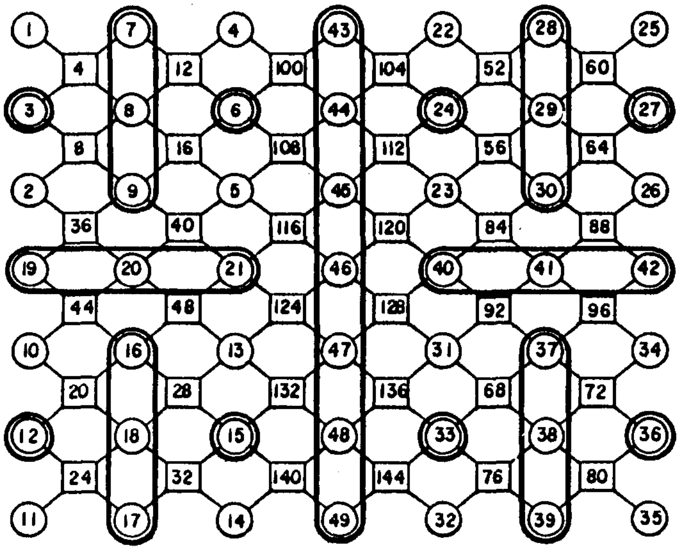

FIG. 7. Example of width-1 nested-dissection column and row orderings.

7, where circles and boxes correspond to column and row vertices respectively.

One way to obtain the decomposition is to use Givens transformations to annihilate the nonzeros in the lower trapezoidal portion of A (and the subsequent reduced matrices) in a column-by-column manner. Let $A_0 = A$, and denote the partially reduced matrix obtained after step i by A_i, $i = 1, 2, \ldots, k^2$. The bipartite-graph model describes the structures of the sequence $\{A_0, A_1, A_2, \ldots, A_{k^2}\}$. See [7] for details. Let $x_i \in S_h$, and consider annihilating the nonzeros below the diagonal element in column i of the reduced matrix A_{i-1}. First note that S_h, which is a level-h separator, is obtained when we dissect a subgrid on the hth level of recursion. Applying Lemma 3.1, one can see that the subgrid will be approximately $2^{-\lfloor h/2 \rfloor}k \times 2^{-\lfloor h/2 \rfloor}k$ or $2^{-\lfloor h/2 \rfloor + 1}k \times 2^{-\lfloor h/2 \rfloor}k$, depending on whether h is odd or even. Moreover, the subgrid may be surrounded by some level-$\bar{h}$ separators $S_{\bar{h}}$, where $\bar{h} < h$. Such a subgrid is said to be *bordered*. The bipartite graph of the subgrid, together with the surrounding separators, is shown in Figure 8.

Note that there are three types of row and column vertices in the bipartite graph associated with the bordered subgrid:

Class 1. This class consists of column vertices in the "boundary" separators (if they exist) and the row vertices that are adjacent to these column

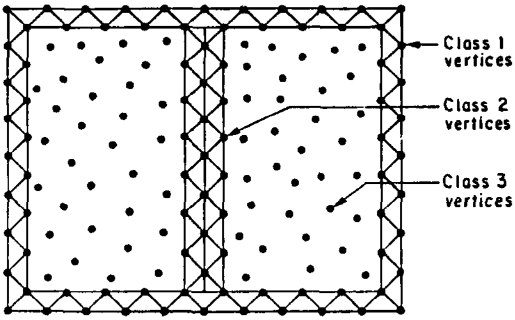

FIG. 8. Bipartite graph of a bordered subgrid.

vertices. In the worst case, there will be approximately $4\times 2^{-\lfloor h/2\rfloor}k$ column vertices and $16\times 2^{-\lfloor h/2\rfloor}k$ row vertices if h is odd, and $6\times 2^{-\lfloor h/2\rfloor}k$ column vertices and $24\times 2^{-\lfloor h/2\rfloor}k$ row vertices if h is even.

Class 2. This class consists of column vertices in S_h and the row vertices that are adjacent to these column vertices. There are approximately $2^{-\lfloor h/2\rfloor}k$ column vertices and $8\times 2^{-\lfloor h/2\rfloor}k$ row vertices.

Class 3. This class consists of the remaining column and row vertices in the bipartite graph of the bordered subgrid.

The important thing to note is that the column and row vertices in class 3 are labeled first, followed by those in class 2, and then those in class 1. This follows from the fact that we are using a width-1 nested-dissection column ordering and its induced row ordering.

The set of nonzeros to be annihilated is given by

$$\mathrm{Adj}_{H_{i-1}}(x_i).$$

Suppose $q_r \in \mathrm{Adj}_{H_{i-1}}(x_i)$, $r > i$. That is, $x_i \in \mathrm{Adj}_{H_{i-1}}(q_r)$. By Theorem 3.1 of [7],

$$x_i \in \mathrm{Reach}_{H(A)}(q_r, \{x_1, x_2, \dots, x_{i-1}, q_1, q_2, \dots, q_{r-1}\}, X(A)). \quad (4.1)$$

Because of the definition of Reach and of the way in which the column and row vertices are labeled, it is easy to see that *only* row vertices in classes 1 and 2 can possibly satisfy (4.1). Hence,

$$|\mathrm{Adj}_{H_{i-1}}(x_i)| \leqslant \alpha_h \times 2^{-\lfloor h/2\rfloor}k,$$

where $\alpha_h = 24$ if h is odd and $\alpha_h = 32$ if h is even.

Now consider $q_r \in \mathrm{Adj}_{H_{i-1}}(x_i)$ again. By Lemma 3.4 of [7], the number of multiplicative operations required to annihilate the corresponding nonzero in A_{i-1} is

$$4\left\{\left|\mathrm{Adj}_{H_i}(q_r)\right|+1\right\}.$$

By Theorem 3.1 of [7] again,

$$\mathrm{Adj}_{H_i}(q_r) \subseteq \mathrm{Reach}_{H(A)}(q_r, \{x_1, x_2, \ldots, x_i, q_1, q_2, \ldots, q_{r-1}\}, X(A)).$$

However, since the column vertices in the "boundary" separators are labeled after those in the subgrid, we have

$$\begin{aligned}&\mathrm{Reach}_{H(A)}(q_r, \{x_1, x_2, \ldots, x_i, q_1, q_2, \ldots, q_{r-1}\}, X(A)),\\ &\quad \subseteq S_h \cup \{\text{class 1 column vertices}\}.\end{aligned}$$

Thus,

$$\left|\mathrm{Adj}_{H_i}(q_r)\right| \leqslant \delta_h \times 2^{-\lfloor h/2 \rfloor} k,$$

where $\delta_h = 5$ if h is odd and $\delta_h = 7$ if h is even. Now the number of multiplicative operations required to annihilate the nonzeros in column i is given by

$$\theta = \sum_{q_r \in \mathrm{Adj}_{H_{i-1}}(x_i)} 4\left\{\left|\mathrm{Adj}_{H_i}(q_r)\right|+1\right\}.$$

Using the inequalities we have derived, θ is given by

$$\begin{aligned}\theta &\leqslant \sum_{q_r \in \mathrm{Adj}_{H_{i-1}}(x_i)} 4\left\{\delta_h \times 2^{-\lfloor h/2 \rfloor} k + 1\right\}\\ &\leqslant 4\left\{\delta_h \times 2^{-\lfloor h/2 \rfloor} k + 1\right\}\left\{\alpha_h \times 2^{-\lfloor h/2 \rfloor} k\right\}\\ &= 4\alpha_h\delta_h \times 2^{-2\lfloor h/2 \rfloor} k^2 + 4\alpha_h \times 2^{-\lfloor h/2 \rfloor} k.\end{aligned}$$

Hence the number of multiplicative operations required to process the columns associated with the column vertices in S_h is given by

$$\theta_h \leqslant \sum_{x_i \in S_h} \left\{4\alpha_h\delta_h \times 2^{-2\lfloor h/2\rfloor}k^2 + 4\alpha_h \times 2^{-\lfloor h/2\rfloor}k\right\}$$

$$\leqslant 2^{-\lfloor h/2\rfloor}k\left\{4\alpha_h\delta_h \times 2^{-2\lfloor h/2\rfloor}k^2 + 4\alpha_h \times 2^{-\lfloor h/2\rfloor}k\right\}$$

$$= 4\alpha_h\delta_h \times 2^{-3\lfloor h/2\rfloor}k^3 + 4\alpha_h \times 2^{-2\lfloor h/2\rfloor}k^2.$$

After some algebraic manipulation and using Lemma 3.1 in Section 3, the total number of multiplicative operations required to reduce A to upper trapezoidal form using Givens transformations is therefore given by

$$\sum_h \theta_h = \sum_{h=1}^{2\log_2 k} \sum_{j=1}^{2^{h-1}} \theta_h = O(k^3).$$

4.2. *Implicit Graph Model*

We assume again that the labeling of the vertices of the symmetric graph $G(M)$ is a width-1 nested-dissection labeling. In [8], we have shown that if S_h is a level-h separator, then $S_h \cup \mathrm{Adj}_{G(M)}(S_h)$ implicitly identifies a set of rows. The row ordering induced by a width-1 nested-dissection labeling can be described as follows. Consider the level-1 separator S_1. We eliminate the rows associated with $S_1 \cup \mathrm{Adj}_{G(M)}(S_1)$ last. The same strategy is then applied recursively to the rows not associated with $S_1 \cup \mathrm{Adj}_{G(M)}(S_1)$.

The following result is useful in deriving the complexity of computing the QR decomposition; it is a consequence of the labeling of the vertices of $G(M)$ and of Lemma 3.1 of [8].

LEMMA 4.1. *Let $x_i \in S_h$. Then*

$$\left|\mathrm{Reach}_{G(A)}(x_i, \{x_1, x_2, \dots, x_{i-1}\})\right| + 1 \leqslant \rho_h \times 2^{-\lfloor h/2\rfloor}k,$$

where $\rho_h = 5$ if h is odd and $\rho_h = 7$ if h is even.

(Note that in this subsection, Reach is applied to the symmetric graph $G(M)$. In the previous subsection, Reach was applied to the bipartite graph $H(A)$. The two Reach operators are not exactly the same, one being the generalization of the other. See [6, 7] for details.)

Another possible way of computing the QR decomposition using Givens transformations is described in [5]. Let $R^0 = O$. Then a sequence of $k^2 \times k^2$ upper triangular matrices $\{R^0, R^1, R^2, \ldots, R^{4(k-1)^2}\}$ is computed, where R^j is obtained from R^{j-1} by rotating in the jth row of A. The implicit graph model describes the structures of the upper triangular matrices and the row-elimination process. See [8] for details.

Consider any row (say row t) associated with $S_h \cup \mathrm{Adj}_{G(A)}(S_h)$. Recall from [8] that the corresponding rotation sequence will involve, in the worst case, vertices in exactly one level-j separator, $h \leqslant j \leqslant 2\log_2 k$. Denote the rotation sequence by Ξ, and let ω_i^t be the number of nonzeros in row i of R^t. Thus, using Lemma 4.1 and the results in [8], the number of multiplicative operations required to eliminate this row is given by

$$\begin{aligned}
\theta &= \sum_{s \in \Xi} \omega_s^t \\
&\leqslant \sum_{j=h}^{2\log_2 k} \sum_{x_i \in S_j} 4\Big\{ \big|\mathrm{Reach}_{G(A)}(x_i, \{x_1, x_2, \ldots, x_{i-1}\})\big| + 1 \Big\} \\
&\leqslant \sum_{j=h}^{2\log_2 k} \sum_{x \in S_j} 4 \times \rho_j \times 2^{-\lfloor j/2 \rfloor} k \\
&= \sum_{j=h}^{2\log_2 k} 4 \times 2^{-\lfloor j/2 \rfloor} k \Big\{ \rho_j \times 2^{-\lfloor j/2 \rfloor} k \Big\} \\
&= \sum_{j=h}^{2\log_2 k} 4 \times \rho_j \times 2^{-2\lfloor j/2 \rfloor} k^2.
\end{aligned}$$

After some algebraic manipulation, we obtain

$$\theta \leqslant 4 \times \sigma_h \times 2^{-h} k^2,$$

where $\sigma_h = 18$ if h is odd and $\sigma_h = 13$ if h is even. Since there are approximately $8|S_h|$ rows associated with $S_h \cup \mathrm{Adj}_{G(A)}(S_h)$, the number of multiplicative operations required to eliminate all these rows will be bounded by

$$\theta_h = 8 \times 2^{-\lfloor h/2 \rfloor} k \big\{ 4 \times \sigma_h \times 2^{-h} k^2 \big\} = 32 \times \sigma_h 2^{-h} 2^{-\lfloor h/2 \rfloor} k^3.$$

Applying Lemma 3.1, the overall number of multiplicative operations required for the computation is then bounded by

$$\sum_h \theta_h = \sum_{h=1}^{2\log_2 k} \sum_{j=1}^{2^{h-1}} 32 \times \sigma_h 2^{-h} 2^{-\lfloor h/2 \rfloor} k^3 = \sum_{h=1}^{2\log_2 k} 16 \times \sigma_h 2^{-\lfloor h/2 \rfloor} k^3 = O(k^3).$$

5. A "BAD" ROW ORDERING FOR NESTED-DISSECTION COLUMN ORDERINGS

In this section, we show that, for the model problem, there exists a row ordering such that the number of multiplicative operations required to compute the QR decomposition using rotations is at least $O(k^4)$, even if the column ordering is a (width-1 or width-2) nested dissection ordering. Throughout this section we will use the implicit graph model, since it is simpler to use in this case.

We assume that the labeling of the grid points is a (width-1 or width-2) nested-dissection ordering. Suppose we eliminate the rows "column by column," starting from the left of the grid. Two examples are shown in Figures 9 and 10 (the numbers inside the small squares refer to the order in which the rows in the squares are processed).

Consider the small squares on the right of the level-1 separator, and partition them by columns. Let D denote the small squares that do not

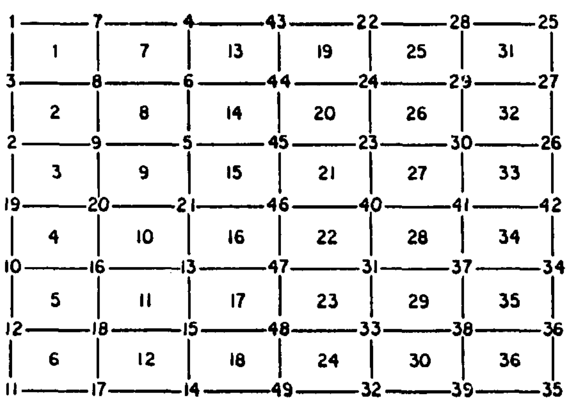

FIG. 9. Example of a bad row ordering for width-1 nested-dissection column ordering.

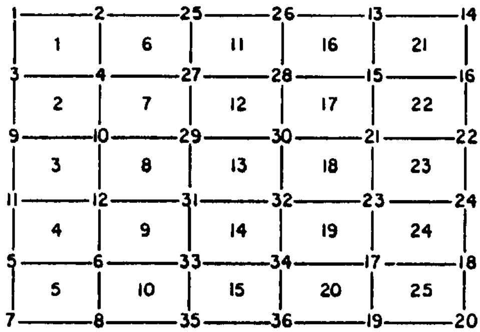

FIG. 10. Example of a bad row ordering for width-2 nested-dissection column ordering.

involve any grid points of the level-1 separator. This set of small squares is shaded in Figure 11. Note that when we start eliminating the rows in the small squares in D, all the rows on the left have been eliminated. Denote the upper triangular matrix obtained at this point by $\bar{R}$. Also denote the union of the graphs of those rows by $\bar{G} = (\bar{X}, \bar{E})$. Note that the set C (see Figure 11) is a subset of $\bar{X}$. Furthermore, for $x_i, x_j \in C$ with $j > i$, $x_j \in \text{Reach}_{\bar{G}}(x_i, \bar{S}_i)$, where $\bar{S}_i = \{x_l \in \bar{X} | l < i\}$. It then follows from Lemma 3.4 of [8] that there is a dense upper triangular submatrix, say $\hat{R}$, in $\bar{R}$ whose order is k. This submatrix corresponds to the unknowns at the grid points in C.

Now consider eliminating the rows in the small squares in D. For each small square, the rotation sequence will include grid points in C. This follows from Lemma 3.6 of [8] and from the fact that the ordering of the grid points is a nested-dissection ordering. That is, for the rows associated with the small squares in D, their elimination will involve the nonzeros of $\hat{R}$. Note that there

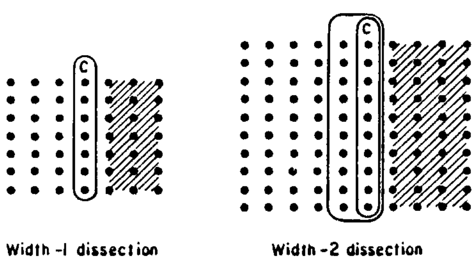

FIG. 11. Set of small squares on the right of the level-1 separator.

are $O(k^2)$ small squares in D and there are $O(k^2)$ nonzeros in $\hat{R}$. Thus, the number of multiplicative operations required to eliminate all the rows in D is at least $O(k^4)$.

6. CONCLUDING REMARKS

In [9], it was shown that, given a sparse rectangular matrix A, if the column labeling is a width-2 nested-dissection labeling, then a good row ordering can be obtained. Furthermore, for the $k \times k$ grid model problem, the number of multiplicative operations required to obtain the orthogonal decomposition using rotations is $O(k^3)$ if a width-2 nested-dissection column ordering and its induced row ordering are used. In [7, 8], we have shown that a good row ordering can also be obtained if a width-1 nested-dissection column ordering is used. In this paper, we have analyzed the model problem using both the bipartite-graph model and implicit graph model and have derived crude bounds on the number of multiplicative operations required to compute the QR decomposition using rotations. The results indicate that the number of multiplicative operations required in the computation is also $O(k^3)$ as long as width-1 nested-dissection column ordering and its induced row ordering are used. In fact, for the model problem, more careful (and tedious) analysis shows that the operation count is smaller for width-1 nested-dissection orderings than for width-2 nested-dissection orderings. In [10], it was shown that the number of multiplicative operations required in the orthogonal reduction is $\frac{1226}{21}k^3 + O(k^2 \log_2 k)$ for width-2 nested-dissection orderings and $\frac{1419}{56}k^3 + O(k^2 \log_2 k)$ for width-1 nested-dissection orderings. These bounds were obtained using the implicit graph model. Numerical experiments have confirmed that the computation using a width-1 nested-dissection column ordering and its induced row ordering in general takes less execution time than that using a width-2 nested-dissection column ordering and its induced row ordering [10].

We have also demonstrated in this paper that for either a width-1 or a width-2 nested-dissection column ordering, there exists a row ordering for which the cost of the computation is at least $O(k^4)$. Thus the problem of finding good row orderings is important and worth studying.

REFERENCES

1 J. H. Argyris and O. E. Bronlund, The natural factor formulation of the stiffness matrix displacement method, *Comput. Methods Appl. Mech. Engrg.* 5:97–119 (1975).

2 J. H. Argyris, T. L. Johnsen, and H. P. Mlejnek, On the natural factor in nonlinear analysis, *Comput. Methods Appl. Mech. Engrg.* 15:389–406 (1978).
3 I. S. Duff, Pivot selection and row ordering in Givens reduction on sparse matrices, *Computing* 13:239–248 (1974).
4 W. M. Gentleman, Row elimination for solving sparse linear systems and least squares problems, in *Proceedings of the 1975 Dundee Conference on Numerical Analysis* (G. A. Watson, Ed.), Lecture Notes in Mathematics 506, Springer, 1975, pp. 122–133.
5 J. A. George and M. T. Heath, Solution of sparse linear least squares problems using Givens rotations, *Linear Algebra Appl.* 34:69–83 (1980).
6 J. A. George and J. W. H. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, N.J., 1981.
7 J. A. George, J. W. H. Liu, and E. G. Y. Ng, Row-ordering schemes for sparse Givens transformations. I. Bipartite graph model, *Linear Algebra Appl.* 61:55–81 (1984).
8 J. A. George, J. W. H. Liu, and E. G. Y. Ng, Row-ordering schemes for sparse Givens transformations, II. Implicit graph model, *Linear Algebra Appl.* 75:203–223 (1986).
9 J. A. George and E. G. Y. Ng, On row and column orderings for sparse least squares problems, *SIAM J. Numer. Anal.* 20:326–344 (1983).
10 E. G. Y. Ng, Row elimination in sparse matrices using rotations, Doctoral Dissertation, Research report CS-83-01, Dept. of Computer Science, Univ. of Waterloo, 1983.

Received 23 September 1983; revised 15 February 1985